

Informe Técnico

Nombre: Gestión de campeonatos de béisbol

Autores:

Ariadna Velázquez Rey C311

Lía López Rosales C312

Carlos Daniel Largacha Leal C312

Gabriel Andrés Pla Lasa C311

Raidel Miguel Cabellud Lizaso C311

Diccionario de datos

Tabla: User

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del usuario.	Primary Key
email	EmailField	Correo electrónico del usuario.	Único, máximo 255 caracteres.
rol_id	ForeignKey (Rol)	Rol asignado al usuario.	No nulo, relación con la tabla Rol.
TD_id	ForeignKey (TechnicalDirector)	Director técnico asociado al usuario.	Nulo permitido, relación con TechnicalDirector .
password	CharField	Contraseña del usuario.	Máximo 128 caracteres.

Restricciones adicionales:

- El campo TD_id no puede ser nulo si el rol_id es "Director Técnico".

Tabla: Rol

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del rol.	Primary Key
type	CharField	Tipo de rol (Director Técnico, Admin, Usuario).	Máximo 50 caracteres, opciones predefinidas.

Tabla: TechnicalDirector

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del director técnico.	Primary Key
direction_team	ForeignKey (DirectionTeam)	Equipo de dirección asociado.	Nulo permitido, relación con DirectionTeam .
W_id	ForeignKey (Worker)	Trabajador asociado al director técnico.	No nulo, relación con Worker .

Restricciones adicionales:

- Clave primaria compuesta: W_id e id.

Tabla: Worker

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del trabajador.	Primary Key
P_id	ForeignKey (Person)	Persona asociada al trabajador.	No nulo, relación con Person .
DT_id	ForeignKey (DirectionTeam)	Equipo de dirección asociado.	Nulo permitido, relación con DirectionTeam .

Restricciones adicionales:

- Clave primaria compuesta: P_id e id.

Tabla: DirectionTeam

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del equipo de dirección.	Primary Key
Team_id	ForeignKey (Team)	Equipo asociado al equipo de dirección.	No nulo, relación con Team .

Tabla: Team

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del equipo.	Primary Key
name	CharField	Nombre del equipo.	Máximo 100 caracteres.
color	CharField	Color del equipo.	Máximo 50 caracteres.
initials	CharField	Iniciales del equipo.	Máximo 10 caracteres.
representative_entity	CharField	Entidad representativa del equipo.	Máximo 100 caracteres.

Tabla: Person

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único de la persona.	Primary Key
CI	IntegerField	Cédula de identidad de la persona.	Único.
age	IntegerField	Edad de la persona.	
name	CharField	Nombre de la persona.	Máximo 100 caracteres.
lastname	CharField	Apellido de la persona.	Máximo 100 caracteres.

Tabla: Position

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único de la posición.	Primary Key
name	CharField	Nombre de la posición.	Máximo 100 caracteres.

Tabla: BaseballPlayer

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del pelotero.	Primary Key
P_id	ForeignKey (Person)	Persona asociada al pelotero.	No nulo, relación con Person .
batting_average	FloatField	Promedio de bateo del pelotero.	
years_of_experience	IntegerField	Años de experiencia del pelotero.	
pitcher	ForeignKey (Pitcher)	Lanzador asociado al pelotero.	Nulo permitido, relación con Pitcher .

Tabla: Pitcher

Campo	Tipo de Dato	Descripción	Restricciones
P_id	ForeignKey (BaseballPlayer)	Pelotero asociado al lanzador.	No nulo, relación con BaseballPlayer .
dominant_hand	CharField	Mano dominante del lanzador.	Opciones: 'izquierda', 'derecha', 'ambas'.
No_games_won	PositiveIntegerField	Número de juegos ganados.	
No_games_lost	PositiveIntegerField	Número de juegos perdidos.	
running_average	PositiveIntegerField	Promedio de carreras permitidas.	

Tabla: BPParticipation

Campo	Tipo de Dato	Descripción	Restricciones
BP_id	ForeignKey (BaseballPlayer)	Pelotero que participa en la serie.	No nulo, relación con BaseballPlayer .
series	ForeignKey (Series)	Serie en la que participa el pelotero.	No nulo, relación con Series .
team_id	ForeignKey (Team)	Equipo al que pertenece el pelotero.	No nulo, relación con Team .

Restricciones adicionales:

- Clave primaria compuesta: BP_id y series.

Tabla: LineUp

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único de la alineación.	Primary Key
team_id	ForeignKey (Team)	Equipo asociado a la alineación.	No nulo, relación con Team.

Restricciones adicionales:

- Clave primaria compuesta: id y team_id.

Tabla: PlayerInLineUp

Campo	Tipo de Dato	Descripción	Restricciones
line_up	ForeignKey (LineUp)	Alineación a la que pertenece el jugador.	No nulo, relación con LineUp.
player_in_position	ForeignKey (PlayerInPosition)	Jugador en una posición específica.	No nulo, relación con PlayerInPosition.

Restricciones adicionales:

- Clave primaria compuesta: line_up y player_in_position.

Tabla: PlayerInPosition

Campo	Tipo de Dato	Descripción	Restricciones
BP_id	ForeignKey (BaseballPlayer)	Pelotero en una posición.	No nulo, relación con BaseballPlayer.
position	ForeignKey (Position)	Posición del pelotero.	No nulo, relación con Position.
effectiveness	FloatField	Efectividad del pelotero en la posición.	

Restricciones adicionales:

- Clave primaria compuesta: BP_id y position.

Tabla: Season

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único de la temporada.	Primary Key
name	CharField	Nombre de la temporada.	

Tabla: Series

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único de la serie.	Primary Key
season	ForeignKey (Season)	Temporada a la que pertenece la serie.	No nulo, relación con Season.
name	CharField	Nombre de la serie.	Máximo 100 caracteres.
type	CharField	Tipo de serie.	Máximo 50 caracteres.
init_date	DateTimeField	Fecha de inicio de la serie.	
end_date	DateTimeField	Fecha de finalización de la serie.	

Restricciones adicionales:

- Clave primaria compuesta: `season` e `id`.
- `init_date` debe ser anterior a `end_date`.

Tabla: Game

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del juego.	Primary Key
local	ForeignKey (TeamOnTheField)	Equipo local.	No nulo, relación con TeamOnTheField .
date	DateTimeField	Fecha del juego.	
rival	ForeignKey (TeamOnTheField)	Equipo rival.	No nulo, relación con TeamOnTheField .
series	ForeignKey (Series)	Serie a la que pertenece el juego.	No nulo, relación con Series .
score	ForeignKey (Score)	Puntuación del juego.	Nulo permitido, relación con Score .

Restricciones adicionales:

- Clave primaria compuesta: `local` y `date`.
- `local` no puede ser igual a `rival`.

Tabla: Score

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del marcador.	Primary Key
winner	ForeignKey (Team)	Equipo ganador.	No nulo, relación con Team .
loser	ForeignKey (Team)	Equipo perdedor.	No nulo, relación con Team .
w_points	PositiveIntegerField	Puntos del equipo ganador.	
l_points	PositiveIntegerField	Puntos del equipo perdedor.	

Restricciones adicionales:

- `winner` no puede ser igual a `loser`.
- `w_points` debe ser mayor o igual a `l_points`.

Tabla: PlayerSwap

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del cambio.	Primary Key
old_player	ForeignKey (BaseballPlayer)	Jugador reemplazado.	No nulo, relación con BaseballPlayer .
date	DateTimeField	Fecha del cambio.	
new_player	ForeignKey (BaseballPlayer)	Jugador que entra.	No nulo, relación con BaseballPlayer .
position	ForeignKey (Position)	Posición del cambio.	No nulo, relación con Position .
game_team	ForeignKey (TeamOnTheField)	Equipo en el campo.	No nulo, relación con TeamOnTheField .

Restricciones adicionales:

- Clave primaria compuesta: `old_player` y `date`.

Tabla: StarPlayer

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del jugador estrella.	Primary Key
series	ForeignKey (Series)	Serie en la que es jugador estrella.	No nulo, relación con Series .
position	ForeignKey (Position)	Posición del jugador estrella.	No nulo, relación con Position .
BP_id	ForeignKey (BaseballPlayer)	Pelotero estrella.	No nulo, relación con BaseballPlayer .

Restricciones adicionales:

- Clave primaria compuesta: **series** y **position**.

Tabla: TeamOnTheField

Campo	Tipo de Dato	Descripción	Restricciones
id	AutoField (PK)	Identificador único del equipo en el campo.	Primary Key
lineup_id	ForeignKey (LineUp)	Alineación asociada al equipo en el campo.	No nulo, relación con LineUp .

Esquema con el diseño de la aplicación

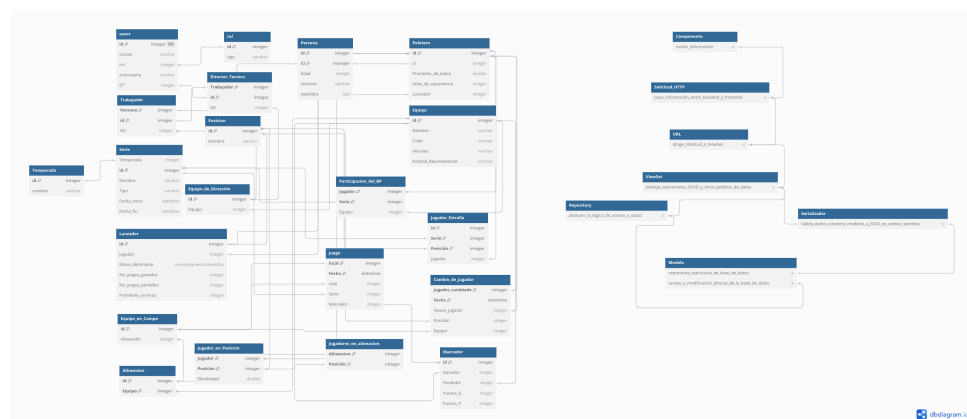


Figure 1: Diagrama de gestión de campeonatos de béisbol

La aplicación sigue un flujo cliente-servidor, donde el frontend (React) y el backend (Django Rest Framework) se comunican a través de solicitudes HTTP. A continuación, se describe el flujo de información y procesamiento en cada capa de la aplicación:

Frontend (React)

El frontend es responsable de la interfaz de usuario (UI) y la interacción con el usuario. Está construido con React, una biblioteca de JavaScript que permite crear interfaces dinámicas y reactivas. El flujo en el frontend es el siguiente:

Interfaz de Usuario (UI)

- React renderiza componentes que representan la interfaz gráfica de la aplicación.
- Los componentes pueden ser formularios, listas, tablas, botones, etc., que permiten al usuario interactuar con la aplicación.

Gestión del Estado

- React utiliza un sistema de estado (por ejemplo, con `useState`, `useReducer` o bibliotecas como Redux) para manejar los datos que se muestran en la interfaz.
- El estado se actualiza dinámicamente en respuesta a las acciones del usuario (por ejemplo, hacer clic en un botón o enviar un formulario).

Comunicación con el Backend

- Cuando el usuario realiza una acción que requiere datos del servidor (por ejemplo, cargar una lista de equipos o enviar un formulario), React realiza una solicitud HTTP (GET, POST, PUT, DELETE) al backend a través de la API REST.
- Para realizar estas solicitudes, se utiliza `fetch` o bibliotecas como Axios.

Renderizado Dinámico

- Una vez que el backend responde con los datos solicitados, React actualiza el estado de la aplicación y re-renderiza los componentes para reflejar los cambios en la interfaz.

Backend (Django Rest Framework)

El backend es responsable de la lógica de negocio, el procesamiento de datos y la comunicación con la base de datos. Está construido con Django Rest Framework (DRF), que proporciona herramientas para crear APIs RESTful de manera eficiente. El flujo en el backend es el siguiente:

Solicitud HTTP

- Cuando el frontend realiza una solicitud HTTP (por ejemplo, una petición GET para obtener datos o POST para enviar datos), Django recibe la solicitud y la procesa.

Enrutamiento (URLs)

- Django utiliza un sistema de enrutamiento (definido en el archivo `urls.py`) para dirigir la solicitud al ViewSet o Vista correspondiente.

Vistas (Viewsets)

- Los ViewSets (o Vistas) son responsables de manejar la lógica de la solicitud. En DRF, los ViewSets permiten realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera simplificada.
- Las vistas interactúan con los serializadores para validar y transformar los datos entrantes o salientes.

Serializadores

Los serializadores en DRF son responsables de:

- Validar los datos enviados por el frontend (por ejemplo, datos de un formulario).
- Convertir los objetos de Django (modelos) en formatos como JSON para enviarlos al frontend.
- Convertir los datos JSON recibidos del frontend en objetos de Django para su procesamiento.

Modelos y Base de Datos

- Los modelos de Django representan la estructura de la base de datos. Cada modelo define una tabla y sus campos.
- Cuando es necesario acceder o modificar datos, las vistas interactúan con los modelos a través del ORM (Object-Relational Mapping) de Django.
- El ORM permite realizar consultas a la base de datos sin escribir SQL directamente, lo que simplifica el manejo de datos.

Repositorios

- Se utilizan repositorios para abstraer la lógica de acceso a datos. Los repositorios actúan como una capa intermedia entre las vistas y los modelos, centralizando las operaciones de la base de datos.

Respuesta HTTP

- Una vez que la vista procesa la solicitud y obtiene los datos necesarios, devuelve una respuesta HTTP al frontend. Esta respuesta suele ser un objeto JSON que contiene los datos solicitados o un mensaje de confirmación.

Flujo Completo de una Solicitud

A continuación, se describe el flujo completo de una solicitud típica en la aplicación:

1. **Usuario realiza una acción en el frontend:**
 - Por ejemplo, el usuario hace clic en un botón para cargar una lista de equipos.
2. **React realiza una solicitud HTTP:**
 - React utiliza Axios o `fetch` para enviar una solicitud GET al endpoint correspondiente en el backend (por ejemplo, `/api/teams/`).
3. **Django recibe la solicitud:**
 - Django dirige la solicitud al `ViewSet` correspondiente a través del archivo `urls.py`.
4. **ViewSet procesa la solicitud:**
 - El `ViewSet` utiliza el serializador para obtener los datos de la base de datos a través del modelo.
 - El serializador convierte los objetos de Django en JSON.
5. **Respuesta al frontend:**
 - El `ViewSet` devuelve una respuesta HTTP con los datos en formato JSON.
6. **React actualiza el estado y la interfaz:**
 - React recibe la respuesta, actualiza el estado de la aplicación y re-renderiza los componentes para mostrar los datos al usuario.

Esquema de las clases definidas

Solución propuesta

Otras informaciones relevantes